

Android App Development

Technical Specification

Abstract

ProMemory è un'app che aiuta a creare e visionare i promemoria degli utenti che la utilizzano

General Technical Data

Choose the features and components included in your app:

- Activity (specify the number of activities in you app)
5 Activity:
 - o AddActivity
 - o MainActivity
 - o ModifyActivity
 - o ReminderActivity
 - o SplashActivity
- Content Provider (specify the type of contents)
SQLite
- Intent (to pass data between activities, to run another app, to run a service.. specify)
Per avviare altre activity e condividere con esse gli id degli oggetti salvati ne database (ReminderActivity, ModifyActivity)
- SQLite database engine (specify the kind of data stored into SQLite db)
Usato per salvare oggetti di tipo Reminder, che definiscono ogni promemoria

Key Features

L'app permette di gestire i promemoria tramite una gestione di preferiti e completati. Ogni promemoria può essere modificato singolarmente in qualsiasi momento.

L'identificazione del completamento di un promemoria avviene in modo dinamico.

Per ogni promemoria viene salvata la data di creazione, un titolo (obbligatorio), un testo che fornisce la descrizione, la data e l'ora di scadenza (che devono essere necessariamente in data odierna o successiva) e un flag se indica se è tra i preferiti oppure no.

La scadenza di un promemoria viene gestita tramite un flag apposito.

L'eliminazione avviene con un tocco prolungato sul promemoria.

App structure overview

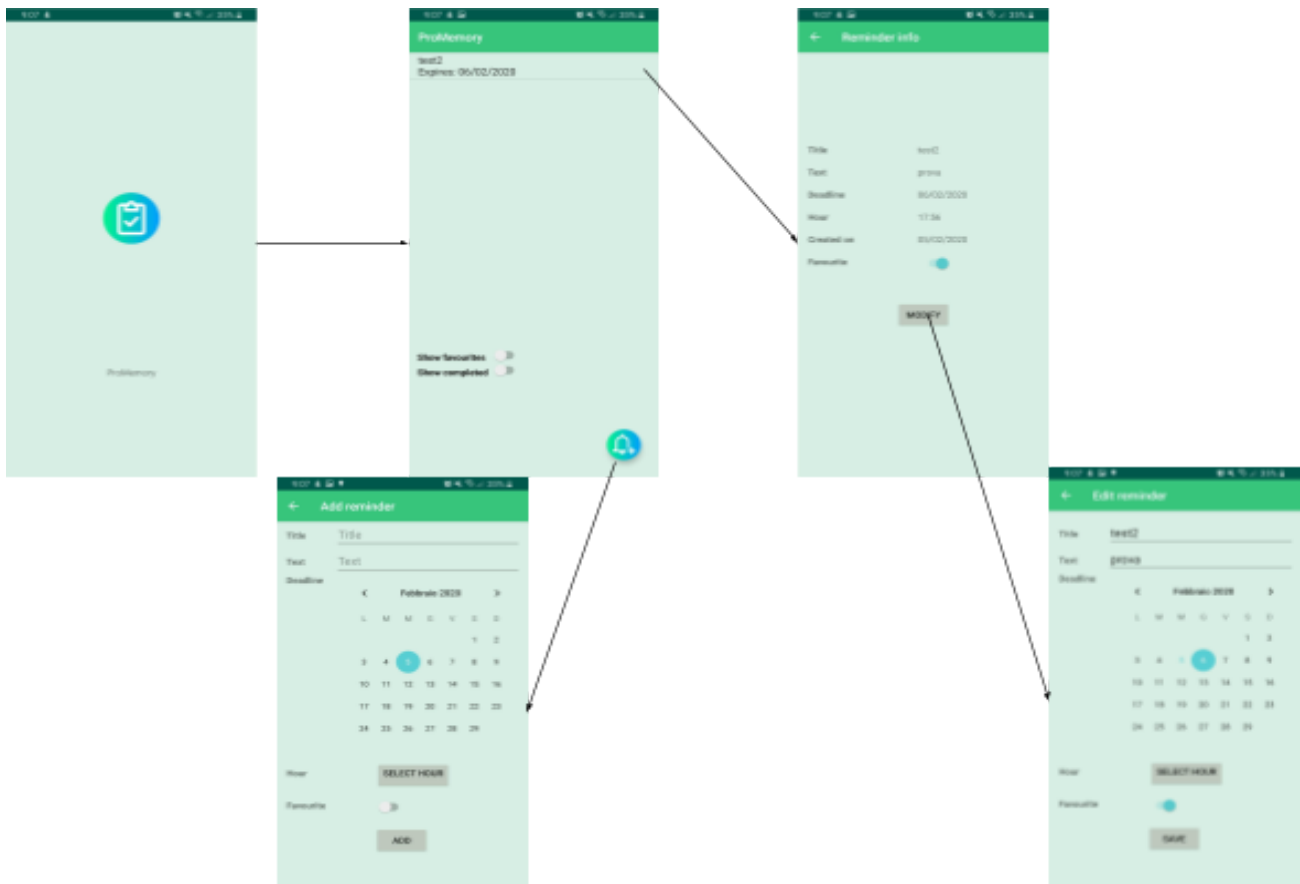
L'app si apre con una splash activity che mostra il logo e il nome per pochi secondi, successivamente si apre l'activity principale che presenta una ListView (che mostra i promemoria richiesti in base all'attivazione degli switch), un floating button per aggiungere un nuovo promemoria e due switch: il primo che, se attivo, mostra solamente i promemoria salvati come preferiti (anche quelli già completati) - il suo stato viene salvato con la chiusura dell'app se questa non viene chiusa del tutto -, il secondo, se attivo, mostra solamente i promemoria già scaduti; l'opzione di default mostra i promemoria non completati, sia preferiti che non.

Tenendo premuto su uno di essi si aprirà un alert dialog che chiederà la conferma dell'eliminazione del promemoria dal database.

Cliccando su uno di essi, si aprirà un'activity che mostrerà i dettagli del promemoria (nome, descrizione, scadenza, creazione, preferito) e presenterà un bottone, che a sua volta aprirà un'altra activity (cliccabile solo se non è completato, altrimenti viene mostrata una snackbar di avviso). L'activity di modifica reperisce tutti i dati del promemoria già esistente e permette di modificarli, inviando le modifiche al database tramite un bottone "SALVA", che chiuderà l'activity e farà tornare l'utente al menù. Cliccando sul floating button verrà aperta un'activity molto simile a quella di modifica, solo che presenterà i campi di inserimento vuoti; l'ora viene scelta tramite un time picker e la data tramite una calendar view.

Ogni activity, eccetto quella principale, presentano un back button sulla parte superiore.

Wireframe



Code fragments

Funzioni per l'aggiornamento della ListView:

```
public static void updateList(){ //function that shows all the reminders that aren't completed
    reminders.clear();
    setCompletedReminders();
    for(Reminder r : db.getAllReminders())
    {
        if(r.getCompleted() == 0) reminders.add(r);
    }
    adapter.notifyDataSetChanged(); //notifies the changes
}

public static void favouriteList(){ //function that shows favourite reminders
    reminders.clear();
    setCompletedReminders();
    for(Reminder r : db.getAllReminders()){
        if(r.getFavourite() == 1 && r.getCompleted() == 0) reminders.add(r); //adds the info only if it is a favourite
        else if(r.getFavourite() == 1 && r.getCompleted() == 1) reminders.add(r);
    }
    adapter.notifyDataSetChanged();
}

public static void completedList(){ //function that shows only completed reminders
    reminders.clear();
    setCompletedReminders();
    for(Reminder r : db.getAllReminders()){
        if(r.getCompleted() == 1) reminders.add(r);
    }
    adapter.notifyDataSetChanged();
}

public static void setCompletedReminders(){ //function that sets the completed flag in all the reminders
    Date current = new Date();
    DateFormat dateFormat = new SimpleDateFormat( pattern: "dd/MM/yyyy");
    String c = dateFormat.format(current);
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(current);
    String hour = calendar.get(Calendar.HOUR_OF_DAY) + ":" + calendar.get(Calendar.MINUTE);
    for(Reminder r : db.getAllReminders()){
        if(AddActivity.compareDate(r.getDeadline(), c) == -1 && AddActivity.compareHour(r.getHour(), hour) == 0){ //if it's the same
            r.setCompleted(1);
            db.updateReminder(r);
        }
        if(AddActivity.compareDate(r.getDeadline(), c) == 0) { //if today is earlier than the deadline then it's completed
            r.setCompleted(1);
            db.updateReminder(r);
        }
    }
}
```

Funzione per condividere il database:

```
public static void start(Context context, MySQLiteHelper db){
    Intent i = new Intent(context, AddActivity.class);
    i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    context.startActivity(i);
    database = db;
}
```

Funzione per confrontare le date:

```
public static int compareDate(String d1, String d2){ //function that compares two dates
    String day1 = d1.substring(0,2);
    String day2 = d2.substring(0,2);
    String month1 = d1.substring(3,5);
    String month2 = d2.substring(3,5);
    String year1 = d1.substring(6,10);
    String year2 = d2.substring(6,10);

    if(Integer.parseInt(year1) < Integer.parseInt(year2)) return 0;
    else if (Integer.parseInt(month1) < Integer.parseInt(month2)) return 0;
    else if (Integer.parseInt(day1) < Integer.parseInt(day2)) return 0;
    else if (Integer.parseInt(year1) == Integer.parseInt(year2) && Integer.parseInt(month1) == Integer.parseInt(month2) && Integer.parseInt(day1) == Integer.parseInt(day2)) return -1;
    return 1;
}
```

Funzione per confrontare le ore:

```
public static int compareHour(String h1, String h2){ //function that compares two hours
    int hour1;
    int minute1;
    int hour2;
    int minute2;

    if (h1.charAt(1) == ':') { //for example 1:54
        hour1 = Integer.parseInt(h1.substring(0, 1));
        minute1 = Integer.parseInt(h1.substring(2, 4));
    }
    if(h1.charAt(2) == ':' && h1.length() == 4){
        hour1 = Integer.parseInt(h1.substring(0, 2));
        minute1 = Integer.parseInt(h1.substring(3,4));
    }
    else {
        hour1 = Integer.parseInt(h1.substring(0, 2));
        minute1 = Integer.parseInt(h1.substring(3, 5));
    }

    if (h2.charAt(1) == ':') {
        hour2 = Integer.parseInt(h2.substring(0, 1));
        minute2 = Integer.parseInt(h2.substring(2, 4));
    }
    if(h2.charAt(2) == ':' && h2.length() == 4){
        hour2 = Integer.parseInt(h2.substring(0, 2));
        minute2 = Integer.parseInt(h2.substring(3,4));
    }
    else {
        hour2 = Integer.parseInt(h2.substring(0, 2));
        minute2 = Integer.parseInt(h2.substring(3, 5));
    }

    if(hour1 < hour2) return 0;
    else if(hour1 == hour2 && minute1 <= minute2) return 0;
    else if(hour1 == hour2 && minute1 == minute2) return -1;
    return 1;
}
```

Development

Target API level:29

Minimum API level: 27

IDE: Android Studio

Man-hours: 60 hh

Problems and difficulties

Le difficoltà principali le ho riscontrate durante il passaggio di dati dal database ad altre activity e per la condivisione di esse con tutte le activity dell'app.

Altre difficoltà sono state trovate durante il controllo della data e dell'ora di un promemoria già esistente (risolte poi con l'implementazione di due funzioni ad-hoc in quanto quelle di sistema non rispondevano alle mie richieste).

Lo sviluppo delle notifiche non è stato ultimato per problemi riguardanti la versione di Android; lo sviluppo di esse è cambiato con le ultime versioni del sistema operativo.

Reported Bugs

Persistono alcuni bug nel caso in cui si aggiunge un promemoria mentre è attivato lo switch per la visualizzazione dei completati.

Se si vuole modificare un promemoria esistente bisogna inserire nuovamente l'ora della scadenza altrimenti non prosegue con il salvataggio

Further development

Il progetto potrebbe essere esteso con l'utilizzo delle notifiche che segnalano quando un promemoria ha raggiunto la scadenza oppure avvisare che sta per scadere con un preavviso di tempo deciso dall'utente

Self-rating

★★★★☆

References

StackOverflow, specialmente per l'integrazione di Snackbar e per risolvere problemi di passaggio dati da database, intent e activity.